

SVG 画像複数人同時編集アプリケーションの開発

大橋 恒平, 松本 章代, Martin J. Dürst

青山学院大学理工学部

1 はじめに

2004 年頃から, Web 2.0 という考え方が注目されている. その定義は未だ正確ではないが, 一般的には Web 2.0 の本質は「ネット上の不特定多数の人々を, 受動的なサービス享受者ではなく能動的な表現者として積極的に巻き込んでいくための技術やサービス」[1] として浸透している. この流れから Web サービスにおいても Ajax などのインタラクティブな技術を駆使した双方向的なものが増加している. また SNS や動画サイトに見られるように, 「体験の共有」が Web サービスのキーワードの一つになっている. 現在では Comet などの発達した技術の出現によって, より同時的なアプリケーションの開発が可能になっている.

そこで我々は, 画像編集という体験を共有できるようなアプリケーションの開発を目指した. それに適した SVG フォーマットを利用して複数人同時編集を可能にするアプリケーションを本研究で実装する.

2 SVG (Scalable Vector Graphics)

2.1 利点

SVG [2] を利用する利点の一つとして形を保ったまま拡大縮小が可能ということが挙げられる. また XML をベースにしているためいくつかの既存の技術をそのまま利用できる. Web での利用に適した画像形式である. DOM [3] や CSS を用いることが可能なので, グラフィックを構造的にとらえることができる. また JavaScript を利用できる. よりインタラクティブな表現が可能になるなど, 拡張性は高い.

2.2 現状と問題点

SVG を利用したライブラリや Web アプリケーションは現在いくつか存在する. しかし, その多くは SVG をプログラムから扱うためのものや, SVG 文書を GUI で編集するためのものであり, SVG の特徴である Web 技術との親和性を活かしたアプリケーションは少ない. そこで我々は現在注目を浴びている Ajax と連携をとったアプリケーションを本研究で実装する.

3 アプリケーションの設計

3.1 概要

本研究では SVG における最大の特徴の一つである, Web で使われている技術との親和性が高いということを利用して, 複数人同時編集を実現する. 今回提案するアプリケーションにおける処理の流れを説明する. まず, あるユーザがブラウザ上で SVG 画像を編集する. このユーザのクライアントを送信クライアントと定義する. 送信クライアントは編集と同時に, その編集内容をサーバに送信する. サーバ側では編集を保存データに適用すると共にすべてのクライアントに編集内容を送信する. これらのクライアントを受信クライアントと定義する. 受信クライアントがサーバから送信された編集内容をそれぞれのブラウザ上に適用する. この一連の処理によって画像編集という体験を複数人で共有することが可能になる. 二つのブラウザが同期している様子を図 1 に例示する.

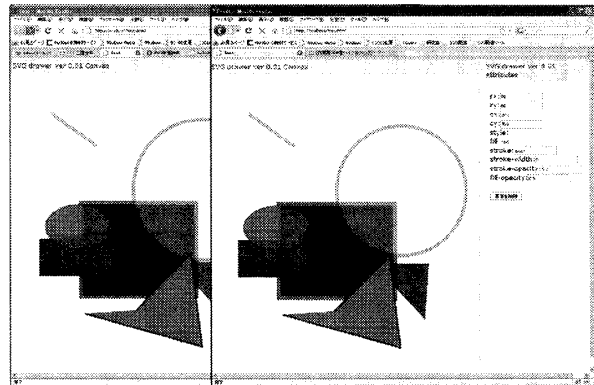


図 1: ブラウザの同期例

3.2 画像編集の同期方法

クライアント側とサーバ側の画像編集の同期にはいくつかの方法が考えられる. まず SVG 文書全体をデータとして通信する方法である. これは文書が複雑になるにつれてデータ量が膨大になる為使用しない. 次に変更による SVG 文書の差分を算出して送信し, サーバ, 受信クライアントそれぞれで差分を適用する方法がある. しかしこの方法では, クライアント側で XML 差分取得, 適用のスクリプトを一から作成しなければならない. 次に変更を加えた際の DOM 上での処理内容 (要素の挿入や属性値の設定など) をデータとして送信し, サーバと受信クライアントそれぞれで実行す

The Development of A Simultaneous Editing Application for SVG
 Kouhei OHASHI, Akiyo MATSUMOTO, Martin J. DÜRST
 Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University
 5-10-1 Fuchinobe, Sagami-hara, Kanagawa 229-8558, Japan
 kouhei@sw.it.aoyama.ac.jp, {akiyo, duerst}@it.aoyama.ac.jp

る方法である。今回のアプリケーションではこの方法で実装する。

処理内容を送信するためには SOAP [4] や XML-RPC [5] が存在する。今回提案するアプリケーションでは通信データ量を考慮して処理内容の単純なテキストデータを独自の規則に従って作成する。図 2 で示すようなデータを送信して行ごとにそれに即した処理を行う。また規則として 1 行目には変更を加えるファイル名、最後の行にはどのユーザが変更を行ったかを判別するための時間によるユーザごとの一意の ID を示すものとする。

```
example.svg
createElement(circle)
setAttribute(cx, 100)
setAttribute(cy, 200)
setAttribute(r, 50)
setAttribute(id, 1230709867058)
appendChild
1230709855609
```

図 2: 交換データ形式の例

3.3 SVG 画像の編集

クライアント側での SVG 画像の編集は DOM ツリーへの編集と同様に JavaScript によって容易にできる。SVG 画像では図形を要素として扱うために、図形単位での追加や移動を実装する。同様の理由から、色の変更などの操作も図形単位で行う。要素のドラッグによる移動にはソースが公開されている dragSVG [6] をカスタマイズして利用する。これによって自由に SVG 要素をドラッグして移動することができる。

サーバ側での SVG 画像の編集には Ruby の標準添付ライブラリである REXML [7] を利用する。REXML は Xpath の実装を含んでいるため、検索式によって id による DOM ツリーに存在する一意の要素を取り出すことができる。また検索した要素に対して自由に変更を加えることができる。

JavaScript と REXML の DOM ツリーに対する処理にはそれぞれ同様の内容のメソッドが存在するので処理内容による同期に適している。

3.4 通信手順

図 3 に示す仕組みで一つの更新につき一回サーバ上のファイルを更新する。それと同時に変更を適用したリクエストで送信されたデータを一時ファイルとして保管する。他のユーザが発行している HTTP リクエストによって起動されている CGI プログラムはファイルの最終変更時間から SVG 文書の変更を検知する。変更が検知されるとブラウザ側に変更内容を送信して、ブラウザ側のプログラムでその更新を適用する。

3.4.1 クライアント - サーバ間の通信

本研究では SVG との親和性が高い Ajax を一部に利用した。Ajax (Asynchronous JavaScript + XML) [8] は通信を非同期で行う。これによってページのリロードを必要とせずにクライアントで行われた処理内容のデータを送信できる。

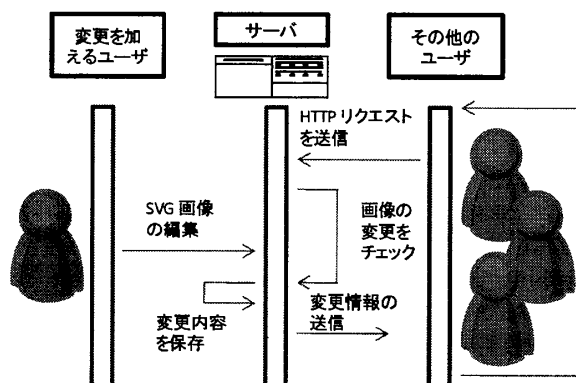


図 3: 通信の様子

3.4.2 サーバ - クライアント間の通信

画像編集の同期には、サーバ側があるユーザから受け取ったメッセージや、ドキュメントに対する変更を即座に他のユーザに対して適用するようなプッシュ型の通信が必要になる。Ajax はこのような機能は持たないため、別の技術が必要になる。これを補うことができる技術が Comet である。Comet [9] では HTTP リクエストをサーバ側で挿入したままにしておくことで疑似的なプッシュを実現する。この技術は Ajax の応用なので SVG にそのまま適用できる。

4 まとめ

本研究で SVG の GUI での編集と Comet との連携による画像編集の体験の共有を提案した。しかし実用に向けて大量ユーザの同時アクセスに対する対処など、改善点は残る。またインターフェースで、より直観的な操作を実現するべきである。

参考文献

- [1] 梅田望夫: ウェブ進化論 本当の大変化はこれから始まる, ちくま新書 (2006).
- [2] Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation, <http://www.w3.org/TR/SVG11/> (2003).
- [3] Document Object Model (DOM) Specification, W3C Recommendation, <http://www.w3.org/DOM/DOMTR> (2004).
- [4] SOAP Version 1.2 Part 0: Primer (Second Edition), W3C Recommendation, <http://www.w3.org/TR/soap12-part0/> (2007).
- [5] XML-RPC Specification, <http://www.xmlrpc.com/> (2004).
- [6] dragSVG, <http://www.codedread.com/dragsvg> (2006).
- [7] Russell, S.: REXML - Home, <http://www.germane-software.com/software/rexml/>.
- [8] Garrett, J. J.: Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/essays/archives/000385.php> (2005).
- [9] Comet, <http://ja.wikipedia.org/wiki/Comet>.